

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Previously Presented) A computer program product, tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause data processing apparatus to:

identify at a client abstraction layer on a server, one or more selection data elements in a client request received at the client abstraction layer, where each selection data element specifies an adapter type, a client type, or data describing the client; and

use the selection data elements to select an adapter at the client abstraction layer to convert communication between an application running on the server and one or more client programs, the adapter being used by the client abstraction layer as an intermediary, the adapter hiding client-specific behavior from the application running on the server, the adapter being designed for use with a particular client program.

2. (Original) The product of claim 1, wherein the instructions to select an adapter comprise instructions to:

perform a multi-stage selection process to select an adapter, the multi-stage selection process comprising:

performing an adapter-request process for selecting an adapter based on the selection data elements that specify the adapter type;

if the adapter-request process fails to select an adapter, performing a client-identification process for selecting an adapter based on the selection data elements that specify the client type; and

if the client-identification process fails to select an adapter, performing a client-description process for selecting an adapter based on the selection data elements that specify data describing the client.

3. (Original) The product of claim 1, wherein the selected adapter makes use of a client capability particular to the client.
4. (Original) The product of claim 3, wherein the client capability comprises the capability to execute instructions in a scripting language.
5. (Original) The product of claim 1, wherein the operation to select an adapter comprises:
 - identifying multiple adapters suitable for communicating with the client; and
 - selecting an adapter from the multiple adapters that makes use of a particular client capability.
6. (Original) The product of claim 1, wherein the operation to select an adapter comprises:
 - identifying multiple adapters suitable for communicating with the client; and
 - selecting an adapter from the multiple adapters that requires the least communication with the client.
7. (Previously Presented) The product of claim 1, wherein the client has multiple client capabilities, and wherein the operation to select an adapter comprises:
 - identifying multiple adapters suitable for communicating with the client; and
 - selecting an adapter from the multiple adapters that is operable to make use of the greatest number of the multiple client capabilities.
8. (Original) The product of claim 1, wherein the adapter is selected from a plurality of adapters stored on a server, the plurality of adapters including one or more of a mobile adapter for a client that comprises a mobile device, an HTML adapter for a client that supports HTML, an XML adapter for a client that supports XML, an RMI adapter for a client that supports RMI, and a JavaScript adapter for a client that supports JavaScript.

9. (Original) The product of claim 2, wherein the client-description process comprises:
using the specification of data describing the client to identify a client capability;
and
wherein the adapter selected to communicate with the client conforms to the client capability.
10. (Original) The product of claim 9, wherein the client capability is a screen size.
11. (Original) The product of claim 2, wherein the client-identification process comprises looking up the specification of the client type in a table.
12. (Original) The product of claim 1, wherein the specification of the client type comprises a specification of a browser and version number.
13. (Previously Presented) A system comprising:
a server having a processor and memory operable to run an application;
a plurality of client-specific adapters, each adapter in the plurality enabling communication between the application on the server and a client; and
a client abstraction layer on the server operable to:
identify one or more selection data elements in a client request received at the client abstraction layer, where each selection data element specifies an adapter type, a client type, or data describing the client; and
use the selection data elements to select an adapter at the client abstraction layer to enable communication between an application running on the server and one or more client programs, the adapter being used by the client abstraction layer as an intermediary, the adapter hiding client-specific behavior from the application running on the server, the adapter being designed for use with a particular client program.
14. (Original) The system of claim 13, wherein the operation to select an adapter comprises:

performing a multi-stage selection process to select an adapter, the multi-stage selection process comprising:

performing an adapter-request process for selecting an adapter based on the selection data elements that specify the adapter type;

if the adapter-request process fails to select an adapter, performing a client-identification process for selecting an adapter based on the selection data elements that specify the client type; and

if the client-identification process fails to select an adapter, performing a client-description process for selecting an adapter based on the selection data elements that specify data describing the client.

15. (Original) The system of claim 13, wherein the adapter is selected from a plurality of adapters stored on a server, the plurality of adapters including one or more of a mobile adapter for a client that comprises a mobile device, an HTML adapter for a client that supports HTML, an XML adapter for a client that supports XML, an RMI adapter for a client that supports RMI, and a JavaScript adapter for a client that supports JavaScript.

16. (Previously Presented) A method comprising:

identifying at a client abstraction layer on a server, one or more selection data elements in a client request received at the client abstraction layer, where each selection data element specifies an adapter type, a client type, or data describing the client; and

using the selection data elements to select an adapter at the client abstraction layer to convert communication between an application running on the server and one or more client programs, the adapter being used by the client abstraction layer as an intermediary, the adapter hiding client-specific behavior from the application running on the server, the adapter being designed for use with a particular client program.

17. (Original) The method of claim 16, wherein selecting an adapter comprises:

performing a multi-stage selection process to select an adapter, the multi-stage selection process comprising:

performing an adapter-request process for selecting an adapter based on the selection data elements that specify the adapter type;

if the adapter-request process fails to select an adapter, performing a client-identification process for selecting an adapter based on the selection data elements that specify the client type; and

if the client-identification process fails to select an adapter, performing a client-description process for selecting an adapter based on the selection data elements that specify data describing the client.

18. (Previously Presented) An apparatus comprising:

means for identifying at a client abstraction layer on a server having a processor and memory, one or more selection data elements in a client request received at the client abstraction layer, where each selection data element specifies an adapter type, a client type, or data describing the client; and

means for using the data elements to elect an adapter at the client abstraction layer to convert communication between an application running on the server and one or more client programs, the adapter being used by the client abstraction layer as an intermediary, the adapter hiding client-specific behavior from the application running on the server, the adapter being designed for use with a particular client program.

19. (Original) The apparatus of claim 18, wherein the means for selecting an adapter comprises:

performing a multi-stage selection process to select an adapter, the multi-stage selection process comprising:

performing an adapter-request process for selecting an adapter based on the selection data elements that specify the adapter type;

if the adapter-request process fails to select an adapter, performing a client-identification process for selecting an adapter based on the selection data elements that specify the client type; and

if the client-identification process fails to select an adapter, performing a client-description process for selecting an adapter based on the selection data elements that specify data describing the client.

20. (Original) The apparatus of claim 18, wherein the adapter is selected from a plurality of adapters stored on a server, the plurality of adapters including one or more of a mobile adapter for a client that comprises a mobile device, an HTML adapter for a client that supports HTML, an XML adapter for a client that supports XML, an RMI adapter for a client that supports RMI, and a JavaScript adapter for a client that supports JavaScript.

21. (Currently Amended) A method of selecting an adapter for converting communication between a plurality of client programs and a server application comprising:

at a client abstraction layer, receiving a connection request from a client program to begin a new connection with the server application, wherein the client abstraction layer hides the client-specific behavior of the client program from the server application, wherein the client abstraction layer is an intermediary between the client program and the server application; and

executing a multi-stage selection process wherein the process comprises:

selecting a process from a plurality of processes based on the connection request;

and

selecting the adapter from a plurality of adapters based on the selected process, wherein each adapter from the plurality of adapters is designed for use with a particular type of client.

22. (Original) The method of claim 21, wherein a first process from the plurality of processes comprises:

receiving an adapter type specified in the connection request; and

selecting the adapter based on the adapter type.

23. (Original) The method of claim 22, wherein a second process from the plurality of processes comprises:

receiving a client type specified in the connection request;

mapping the client type to the adapter, wherein the mapping performs a query in a property file that maps a plurality of client types to a plurality of adapters; and

selecting the adapter corresponding to the client type.

24. (Original) The method of claim 23, wherein a third process from the plurality of processes comprises:

receiving information descriptive of the client's capabilities in the connection request;

receiving a set of client conditions for each adapter belonging to the plurality of adapters, wherein the set of client conditions specify the minimum requirements for using that particular adapter; and

selecting one or more adapters that meet the client conditions.

25. (Original) The method of claim 24, wherein the third process further comprises selecting the adapter from the one or more adapters based on a priority list, wherein the priority list ranks the plurality of adapters according to pre-defined criteria.